Forest Query

Thái Bao Võ Huỳnh

July 2025

1 Forest Queries

Problem: Given an $n \times n$ grid of trees (* characters) and empty spots (.), answer q queries about the number of trees in a given subrectangle.

Constraints:

- $1 \leq n \leq 1000$
- $1 \leq q \leq 2 \cdot 10^5$
- $1 \leq y_1 \leq y_2 \leq n$
- $1 \leq x_1 \leq x_2 \leq n$

Approach: Precompute a 2D prefix sum matrix where each cell contains the total number of trees from $(1, 1)$ to (i, j). Then, each query is answered in constant time using the inclusion-exclusion principle.

Sample Input:

```
4 3
.*..
*.**
**..
****
2 2 3 4
3 1 3 1
1 1 2 2
```

Sample Output:

```
3
1
2
```

Time Complexity:

- Preprocessing: $\mathcal{O}(n^2)$
- Each query: $\mathcal{O}(1)$

2 Christmas Party

Problem: There are n children at a Christmas party, and each of them has brought a gift. The idea is that everybody will get a gift brought by someone else.

In how many ways can the gifts be distributed?

Input: The only input line has an integer n: the number of children.

Output: Print the number of ways modulo $10^9 + 7$.

Constraints:

- $1 \leq n \leq 10^6$

Example:

```
Input:
4
Output:
9
```

Approach: This is a classic derangement problem. The number of valid gift distributions (where no one gets their own gift) is equal to the number of *derangements* of n elements. The recurrence relation is:

$$D(n) = (n-1) \cdot (D(n-1) + D(n-2))$$

With base cases:

$$D(1) = 0, \quad D(2) = 1$$

Use dynamic programming with modulo $10^9 + 7$ to avoid overflow.

Time Complexity: $\mathcal{O}(n)$